Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 119 (2015) 1375 – 1380

Procedia Engineering

www.elsevier.com/locate/procedia

13th Computer Control for Water Industry Conference, CCWI 2015

Application of HADOOP to Store and Process Big Data Gathered from an Urban Water Distribution System

Tomasz Jach[a,*], Ewa Magiera[a], Wojciech Froelich[a]

[a]*Institute of Informatics, University of Silesia, Bedzinska 39, 41-200 Sosnowiec, Poland*

Abstract

Information technology has become an integral part of municipal water distribution systems (WDS). Various types of sensors, e.g., smart water meters, usually work in real-time mode delivering a huge amount of data. Big data must be stored in appropriate databases. Along with the development of data mining tools, the analysis of big data is very important for the management of WDS. Valuation of NoSQL databases for water data is currently in its very early stages. In this paper, the Apache Hadoop platform is investigated with respect to a possible database solution based on NoSQL. We present comparative experiments evaluating the performance of the Hadoop and MySQL databases.

Peer-review under responsibility of the Scientific Committee of CCWI 2015

Keywords: Big data; Hadoop; MapReduce; SQL; water distribution system

1. Introduction

Big Data is the notion used to describe large and complex datasets. Such sets are often subject to different, optimised processing techniques. Their vast spectrum, great volume and complicated structure make ordinary data retrieval and manipulation algorithms obsolete. Event storage, share and transfer are challenging, but accuracy in big data allows greater confidence in decision making and better decisions can lead to greater operational efficiency, cost reductions and reduced risk. Furthermore, big data is often copied, which only increases its size. In fact, the size of these datasets may even reach the level of petabytes. These quantities are too large to store on a single computer, large data must be distributed across multiple computers, often geographically dispersed. The process of dividing data into multiple physical locations by integrating the logical layer creates a distributed file system (DFS) [1]. It allows to share data stored in different places in such a way that the end-user sees it as being stored in a single location.

”Big Data” storage solutions frequently utilise the NoSQL paradigm [2], also called ”Not only SQL”. The meaning of this term is very broad, covering database management systems that deviate from the traditional relational model. Nowadays, there is increasing interest in NoSQL solutions due to their usefulness in processing large amounts of data. In addition, mobile and online systems boost the application of NoSQL techniques. In such systems, data are often semi-structured or even non-structured. One of the principles of NoSQL techniques is their scalability, fast input and

* Corresponding author. Tel.: +48-32-368-97-65

E-mail address: tomasz.jach@us.edu.pl

Peer-review under responsibility of the Scientific Committee of CCWI 2015

doi:10.1016/j.proeng.2015.08.988

output operations, and high system availability. Effective management of such systems is often hampered by the key principles of relational systems such as rigid schema data, ACID properties [3], and JOIN operations.

The idea of non-relational solutions is not new, but their current development is beneficial for technological progress in general with the tremendous amount of heterogeneous data that is currently emerging. As one of the principles of NoSQL is that the solutions based on this paradigm do not have to match predefined schemas, NoSQL solutions do have no predetermined rigid data structure. This is particularly profitable when the stored data are varied or are types other than structural, and it is impossible to predict exactly how much data will flow to the system. A flexible scheme allows one to store a variety of data without needing to create many attributes whose values would remain empty for most objects anyway.

Smart water and smart meters [4] have become a crucial part of smart cities. Big data have been the source for analytical tools embedded in DSS for water demand management at the urban level. Thompson et al. [5] considered the problem of gathering and monitoring water data in real time. The contemporary challenge is mining these data for reliable results that support operational activities in water municipal management. McKenna et al. [6] developed an approach for the classification of demand patterns, and applied this approach to a set of demands collected from smart meters within a single District Metered Area (DMA) of a municipal network. During the last several years, a new generation of systems has been introduced for massive-scale data processing. Systems based on the MapReduce paradigm [7], especially Apache Hadoop [8] have become more popular. They can be used for a huge amount of distributed data and ensure parallel processing. Parallel processing could be applied in big water networks at the region or country region level. Our goal in this paper is to compare and evaluate the traditional SQL approach and HADOOP system in processing data gathered in water distribution systems. There is a dearth of scholarship comparing the performance of different parallel data processing systems. In one of the first works, Pavlo et al. [9] comparatively benchmarked a parallel relational DBMS (i.e., Hadoop [6]) and a column-store parallel DBMS (Vertica [10]). The goal of their work was to evaluate the performance of these three systems in general. The study thus used synthetic data. In our case, we used real data gathered in WDN, and we created queries that are important in the water management domain. Loebman et al. [11] evaluated the performance of a commercial RDBMS and Hadoop on astronomy simulation analysis tasks. In the case of small- to medium-scale clusters, the modern RDBMS offered a powerful platform for organizing data and a satisfactory set of attributes for improving performance. The presented tests also indicated that, overall, a RDBMS outperforms Hadoop for representative queries in the astronomy domain.

2. Integrated Support System for Efficient WATer Usage and resources management

Integrated Support System for Efficient Water Usage and resources management (ISS-EWATUS) is a project implemented by an international consortium. Founded by the 7th Framework Programme, the project proposes several innovative ICT methods to achieve a multi-factor system capable of optimising water management and reducing water usage at the urban and household levels.

At the urban level, the main goal of the ISS-EWATUS is to reduce water leaks within the water distribution system by maintaining water pressure within appropriate bounds. The urban DSS will help water companies identify leaks and suggest emergency actions, assess demands in the medium and long term, and manage the demands through an optimal balance between supply and demand measures. Data collected from water distribution systems will be used to analyse consumption patterns and provide evidence of leaks and trigger alerts; the gathered data will also be used to predict future water consumption based on historical consumption and other pertinent parameters. One location of project validation is the city Sosnowiec, located in the Upper Silesian in Poland. The Regional Water Supply and Sanitation Company in Sosnowiec (RPWiK) is responsible for developing maintaining the water distribution network in the city. The company tightly cooperates with ISS-EWATUS providing invaluable input and expert knowledge.

In 2014 the total length of the water distribution system in Sosnowiec was 580 km, including the supply distribution network, network connections and leased sections. Pipelines are made of polyethylene (PE) 54%, steel 24%, cast iron 20% and polyvinyl chloride (PVC) 1%. The water distribution network has two pressure zones the urban zone and the pumping stations zone. In 2014, the company owned 24 pumping stations that pump water to high-rise blocks of flats (usually higher than 4 floors).

To reduce water losses, the local water distribution company continuously monitors conditions within the network by means of sensors installed at 44 purchase wells and 47 local water meters. Additionally, the company controls

the distribution of water to particular recipients. In 2014, there were approximately 14,000 radio-read water meters installed at households and public use institutions.

The planned DSS at the urban level will based on a huge amount of data regarding pressures and flows. To meet the assumed requirements, as well as ensure the record data in real time, the following equipment has been installed:

1. Dual-chamber pressure regulation valve (PRV) activated and controlled by the current water pressure, it does not need any external power source. It is a very cost-effective and optimal solution for the automation of industrial, public and agricultural water supply systems. Additionally, the PRV can be combined with the electronic drivers that can even more effectively adjust the pressure to highly variable demands;
2. Electronic driver for the (PRV) that is 'Regulo' enables wireless data transmission by GSM/SMS/GPRS at selected time intervals. It is equipped with sensors that control pressure at the inlet and outlet from the valve, and enables the remote regulation of data recordings and pressure;
3. Workstation software 'PMAC Plus' by Technolog, which provides access to collected data;
4. Water meter 'Woltex', which monitors flow near the PRV; and
5. Pressure and flow sensors (Cello recorder) installed at the critical point, it is user-programmable and monitors water flow and pressure, and can record data at intervals between 1 second and 1 hour.

3. Data Gathering process

Water utilities collect vast amounts of data for operating and maintaining WDS. Those data include information about pressure, flows, treatment, distribution, storage and pumping, in addition to water consumption and billing data for water users. In this type of system, the data processed is often semi-structured. To detect and prevent irregularities, these data must be analysed as soon as they appear, preferably in real time. In addition, the system and the continuous development of European Union directives necessitate much more frequent measurements and the collection of large amounts of historical data.

Additionally, to conduct further processing of data, the data gathered from WDS have been supplanted by weather information (rainfall along with minimal, maximal and average temperatures during the day). Incoming data consists of the set of data sequences presented in Figure 1.

```
01120920131312005650 5+098+161+120015
01120920131318005650 6+098+161+120015
01120920131324005650 5+098+161+120015
01120920131330005650 5+098+161+120015
```

Fig. 1. Input data format

The data contain attributes, which are described below, based of the example above:

- 01 - Station ID,
- 12 - day number,
- 09 - month number
- 2013 - year,
- 15 - hour,
- 06 - minutes,
- 00 - seconds,
- 570 - recorded pressure value multiplied by ten,
- 05 - recorded flow rate multiplied by ten,
- +098 - minimum temperature recorded on that day multiplied by ten,
- +161 - the maximum temperature recorded on that day multiplied by ten,
- 120 - average temperature of the day multiplied by ten
- 015 - rainfall for the day multiplied by ten.

3.1. Data usage

The DSS for reducing background leaks in the water delivery system is currently under development. In order to properly meet the requirements, the data gathering process must be done very smoothly. Apart from that, computations have to be made very quickly and online. Both of these limitations cause us to lean more towards using one of the NoSQL database systems. From the preliminary experiments, Apache Hadoop was chosen. The WDS is also interested in an adaptive pricing policy as the economic instrument to induce water-saving behaviour and reduce peaks in water and energy distribution loads. The data from all municipal companies is stored in the same database. As such, the volume of these is enormous. In Sosnowiec, there are more than two dozen points where pressure and water flow are measured. Given the fact, that for each measuring point, the data is transmitted every five minutes, on each day at least seven thousand records are created. After quick calculations, it gives more than two and a half million records per year. These requirements are very close to a popular recent trend in data processing called ”Big Data” [1].

4. Apache HADOOP and NoSQL

Hadoop is an open-source platform developed by the Apache Software Foundation [8] utilising the NoSQL paradigm. It is based on a distributed file system by Google using the MapReduce model [12]. The Hadoop platform is an environment designed for building reliable and scalable parallel systems. They can perform operations on very large files that are stored in a distributed file system. Furthermore, the platform is capable of managing all types of data, especially unstructured ones. The platform consists of the following elements:

- Hadoop Distributed File System (HDFS), which provides high data availability;
- Structure of Hadoop YARN, used for planning tasks and management of clusters;
- Implementation of MapReduce, based on YARN structure, for parallel processing on large datasets; and
- Hadoop Common, a set of services to support the other modules.

An example of a distributed file system is the Hadoop Distributed File System (HDFS), which is part of many major Big Data platforms. HDFS is designed to store large amounts of data distributed among multiple servers. These servers are often organized in co-operation with each other in clusters, i.e., sets of computers creating a dedicated network. The Hadoop file system was adapted to perform tasks in accordance with the MapReduce model, which means that each analysis requires a sequential read stream of a large part of the stored data (often the whole system). Sadly though, the HDFS is not suitable for operation on single records or small groups. The effectiveness of the system in this case is worse than with other systems.

5. Computational experiments

For comparative purposes, several experiments have been performed using Apache HADOOP and MySQL. Both database servers were installed ”as is” without any improvements or performance tweaks. In both cases, sample data of different volumes were inserted using standard methods. The data involved pressure and flow readings from different time periods.In MySQL, all data were stored in two tables (for flow and pressure, respectively). Both tables had three columns: the unique identifier (INTEGER type), date of measurement (TIMESTAMP type) and value of reading (FLOAT). Apache HADOOP was populated analogously but with respect for different storage mechanism. Four tasks were conducted to compare the performance of the two servers considered:

- Finding the maximum value of flow/pressure from the whole dataset (Figure 5 labels MySQL and HADOOP respectively);
- Finding the maximum value of flow/pressure in each month (Figure 5 labels Group MySQL and Group HADOOP respectively);
- Finding the average value of flow/pressure from the whole dataset (Figure 5 labels MySQL and HADOOP respectively);

- Finding the average value of flow/pressure in each month (Figure 5 labels Group MySQL and Group HADOOP respectively).

Table 1. The results of computational experimens.

	MAX		AVG		Group MAX		Group AVG	
No of data	MySQL	HADOOP	MySQL	HADOOP	MySQL	HADOOP	MySQL	HADOOP
800 000	-	5	-	4	-	6	-	4
1 600 000	-	6	-	5	1	7	1	5
3 200 000	-	9	1	7	2	9	2	10
6 400 000	3	12	2	12	4	15	4	14
12 800 000	6	20	5	22	8	25	9	26
25 600 000	10	38	10	37	16	55	16	47

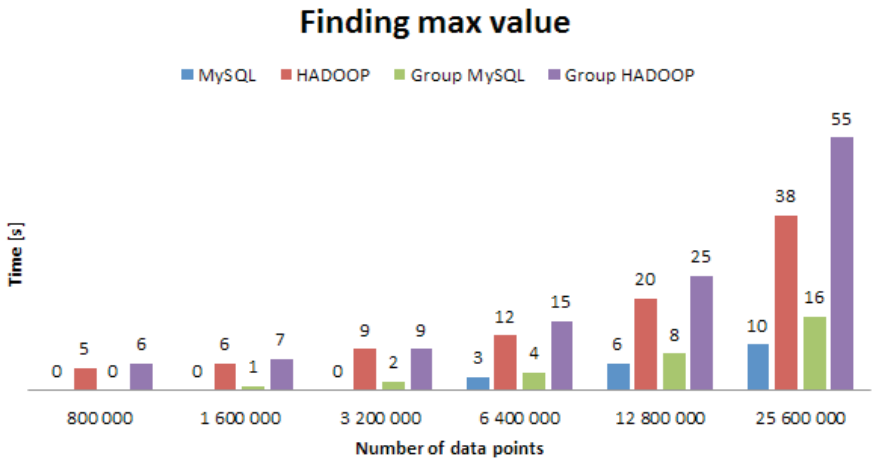

Fig. 2. The results for finding the maximal value

The results are given in Table 5. The symbol '-' denotes the value that is close to 0, lower then the assumed rounding. During each experiment, the time necessary for the query to be fully executed was measured. As can noticed, MySQL greatly outperformed HADOOP. The difference between these two solutions is significant, as the latter system spent nearly four times more time than its competitor.

It became clear that MySQL outperforms the HADOOP. It is worth mentioning that both solutions scale up in the same way. At some point, each test took twice as much time as the previous one (NB: the size of the input data was doubled in each consecutive test).

6. Summary

In this paper we performed experiments revealed that the MySQL database outperform HADOOP. One of the main benefits of HADOOP, is its ability to work in a distributed environment. Due to the nature of the gathering process, which combines data from multiple locations, the geographical distribution of the databases can be an advantage. In addition, given HADOOPs ability to automatically replicate data as well as its security, reliability and overall effectiveness can make it vastly beneficial as well. Further experiments validating other NoSQL solutions in the WDS should be pursued in future research as well. The preliminary experiments also showed that other NoSQL solutions are

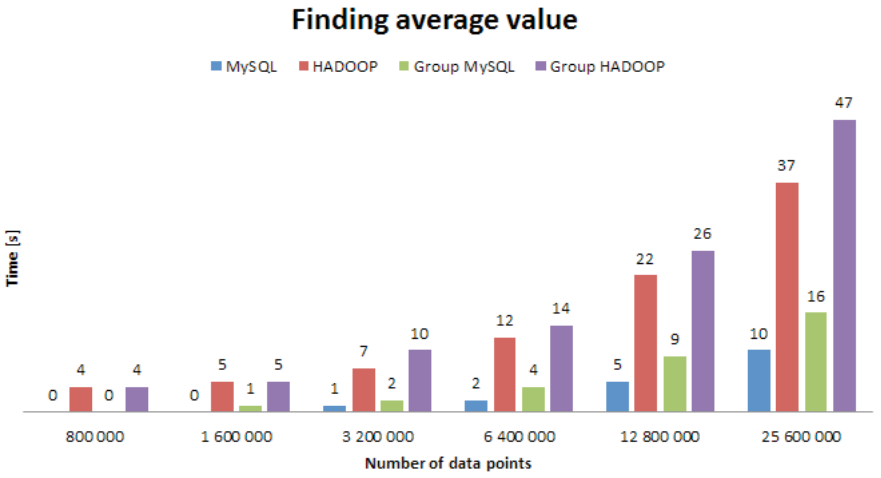

Fig. 3. The results for finding the average value

more efficient (e.g. MongoDB [13]). The authors would like to optimise the performance of these, without losing the auto-mirroring feature. The works will be continued, mainly towards minimising the performance gap of HADOOP.

Acknowledgements

This work is a part of the project *"Integrated Support System for Efficient Water Usage and Resources Management"* founded by the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no [619228].

References

[1] K. Shvachko, H. Kuang, S. Radia, R. Chansler, The hadoop distributed file system, in: Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on, IEEE, 2010, pp. 1–10.
[2] N. Leavitt, Will NoSQL databases live up to their promise?, Computer 43 (2010) 12–14.
[3] M. Keith, Pro JPA 2 (2013).
[4] T. Gurunga, R. Stewart, A. Sharmac, C. Bealda, Smart meters for enhanced water supply network modellingand infrastructure planning (2014).
[5] K. Thompson, R. Kadiyalab, Leveraging big data to improve water system operations (2014).
[6] S. McKenna, F. Fuscoa, B. Ecka, Water demand pattern classification from smart meter data (2013).
[7] H. Yang, A. Dasdan, R.-L. Hsiao, D. S. Parker, Map-reduce-merge: simplified relational data processing on large clusters (2007).
[8] Apache foundation: Hadoop, 2014. URL: `http://hadoop.apache.org/`.
[9] A. Pavlo, E. Paulson, A. Rasin, D. J. Abadi, D. J. DeWitt, S. Madden, M. Stonebraker, A comparison of approaches to large-scale data analysis, in: Proceedings of the 2009 ACM SIGMOD International Conference on Management of data, ACM, 2009, pp. 165–178.
[10] Vertica, inc., 2014. URL: `http://www.vertica.com/`.
[11] S. Loebman, D. Nunley, Y. Kwon, B. Howe, M. Balazinska, J. P. Gardner, Analyzing massive astrophysical datasets: Can Pig/Hadoop or a relational DBMS help?, in: Cluster Computing and Workshops, 2009. CLUSTER'09. IEEE International Conference on, IEEE, 2009, pp. 1–10.
[12] J. Dean, S. Ghemawat, MapReduce: simplified data processing on large clusters (2014).
[13] K. Chodorow, MongoDB: the definitive guide, " O'Reilly Media, Inc.", 2013.